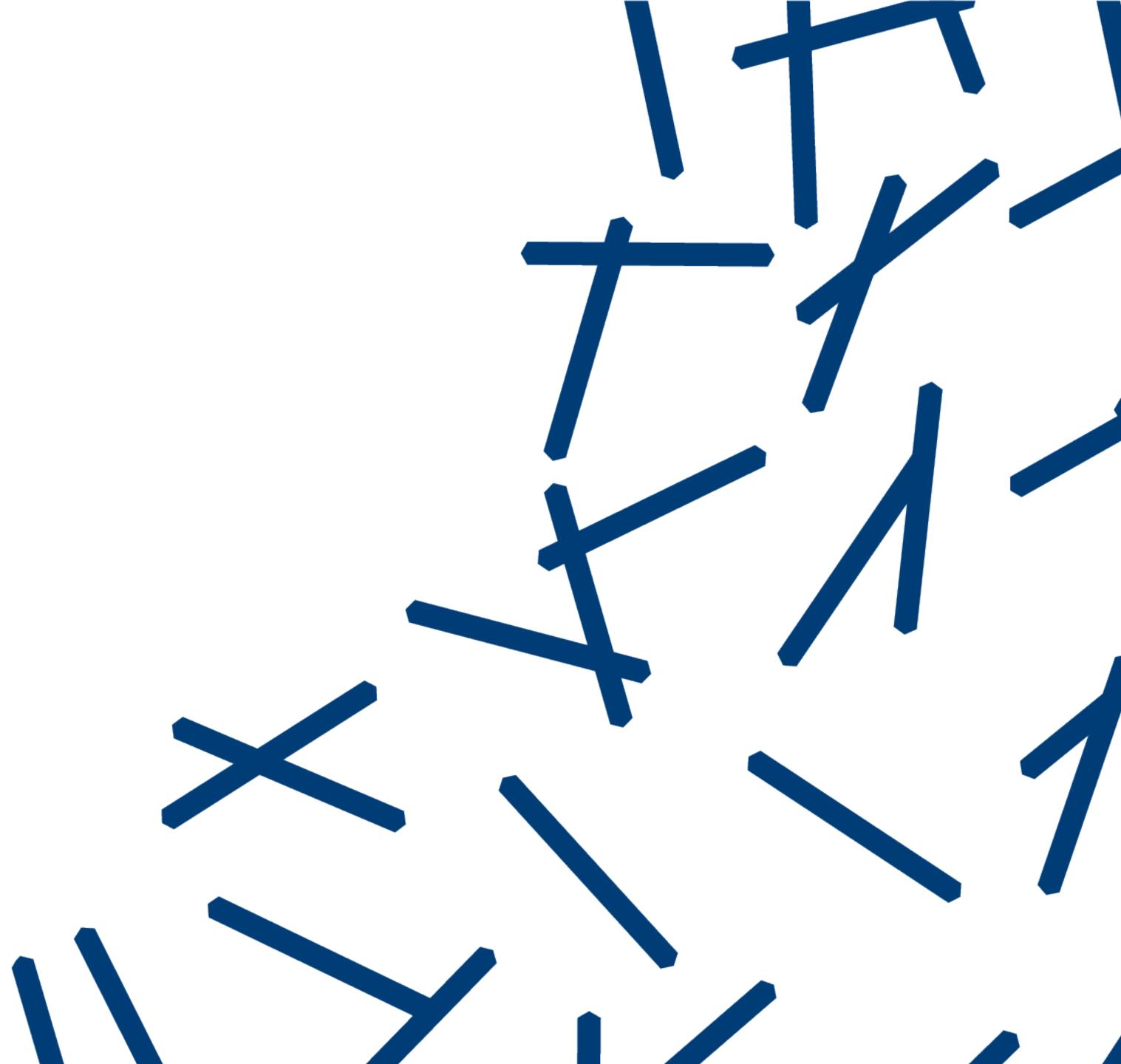
Yale engineering

Non-determinism

Rajit Manohar





Non-deterministic selection

- When a communication is pending on both A and B
 - We can pick either... but we have to make a choice
 - * The **arbitration** problem

Should two courses be judged equal, then the will cannot break the deadlock, all it can do is to suspend judgement until the circumstances change, and the right course of action is clear.

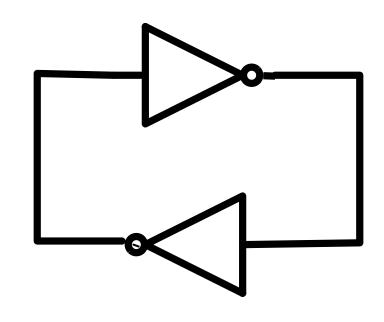
— Jean Buridan, c. 1340





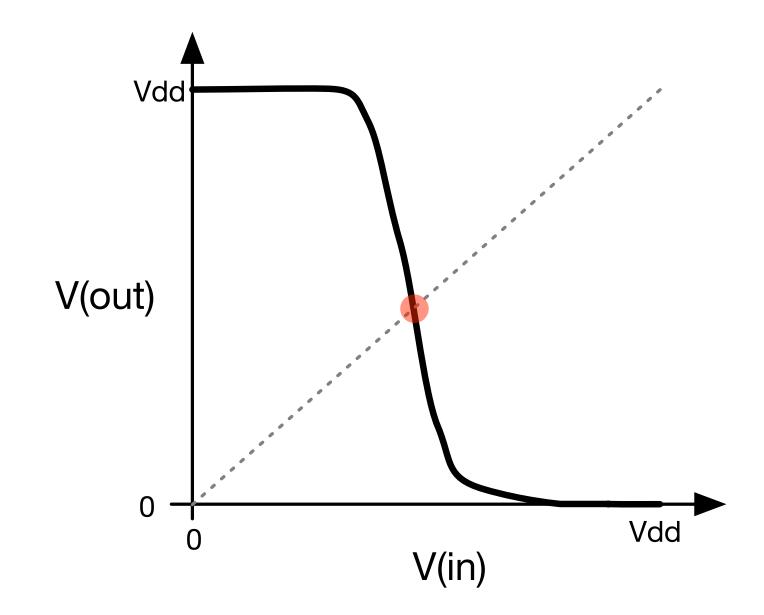
Metastable state

- Unstable equilibrium
 - Inverted pendulum
 - Balancing a pencil on its tip

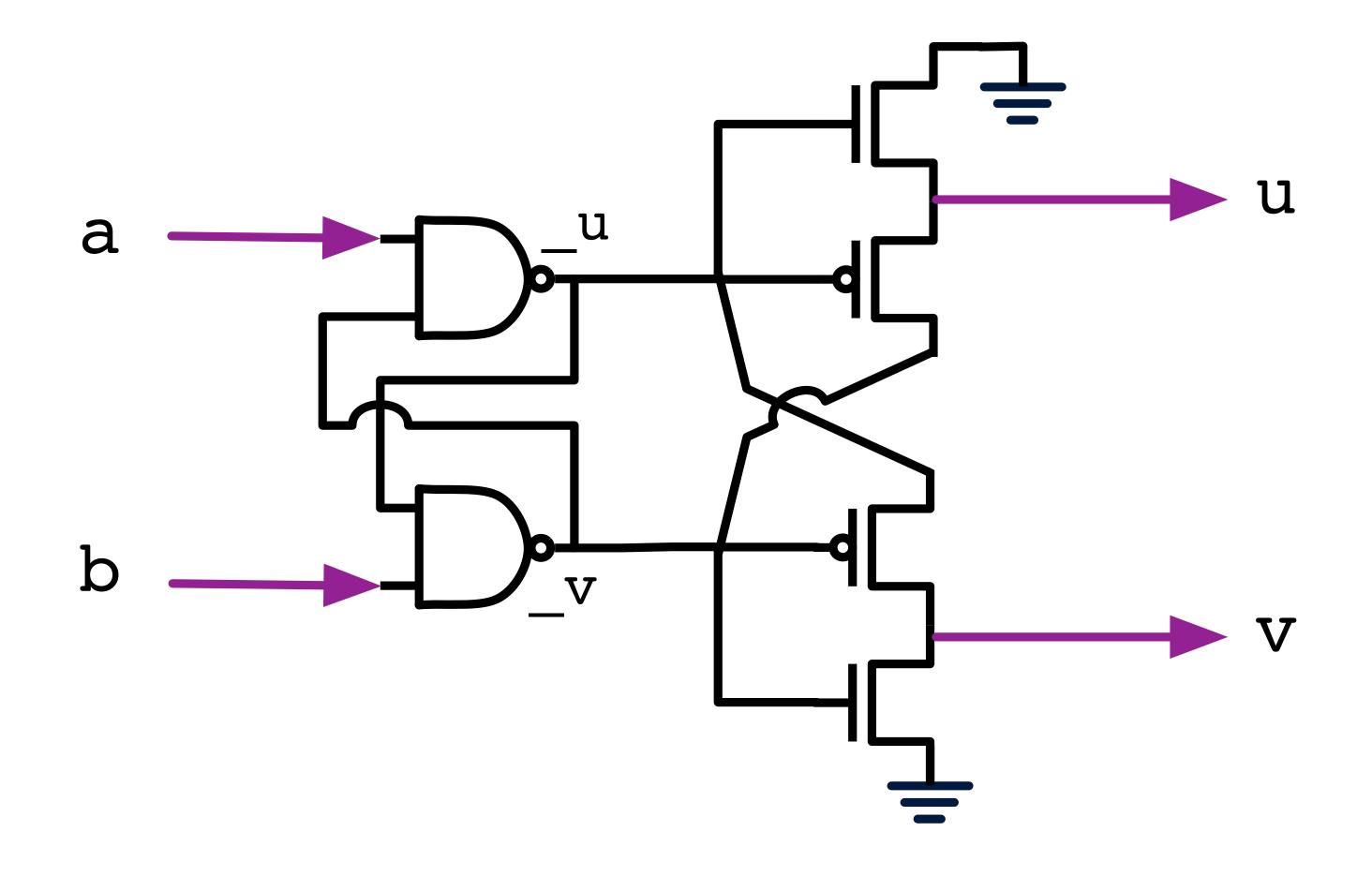




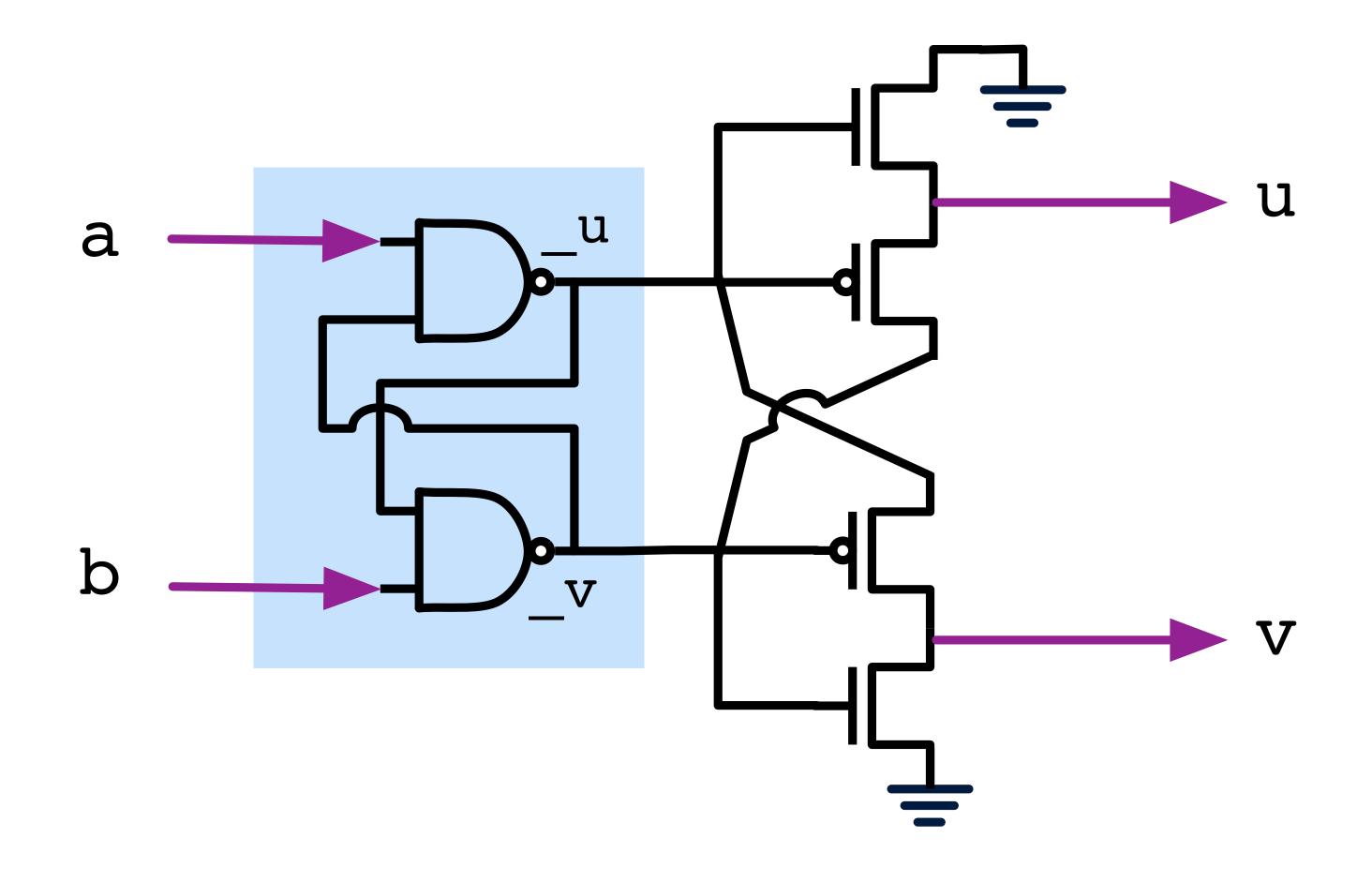
- Stable states: 0/1 and 1/0
- Metastable state: V(out) = V(in)



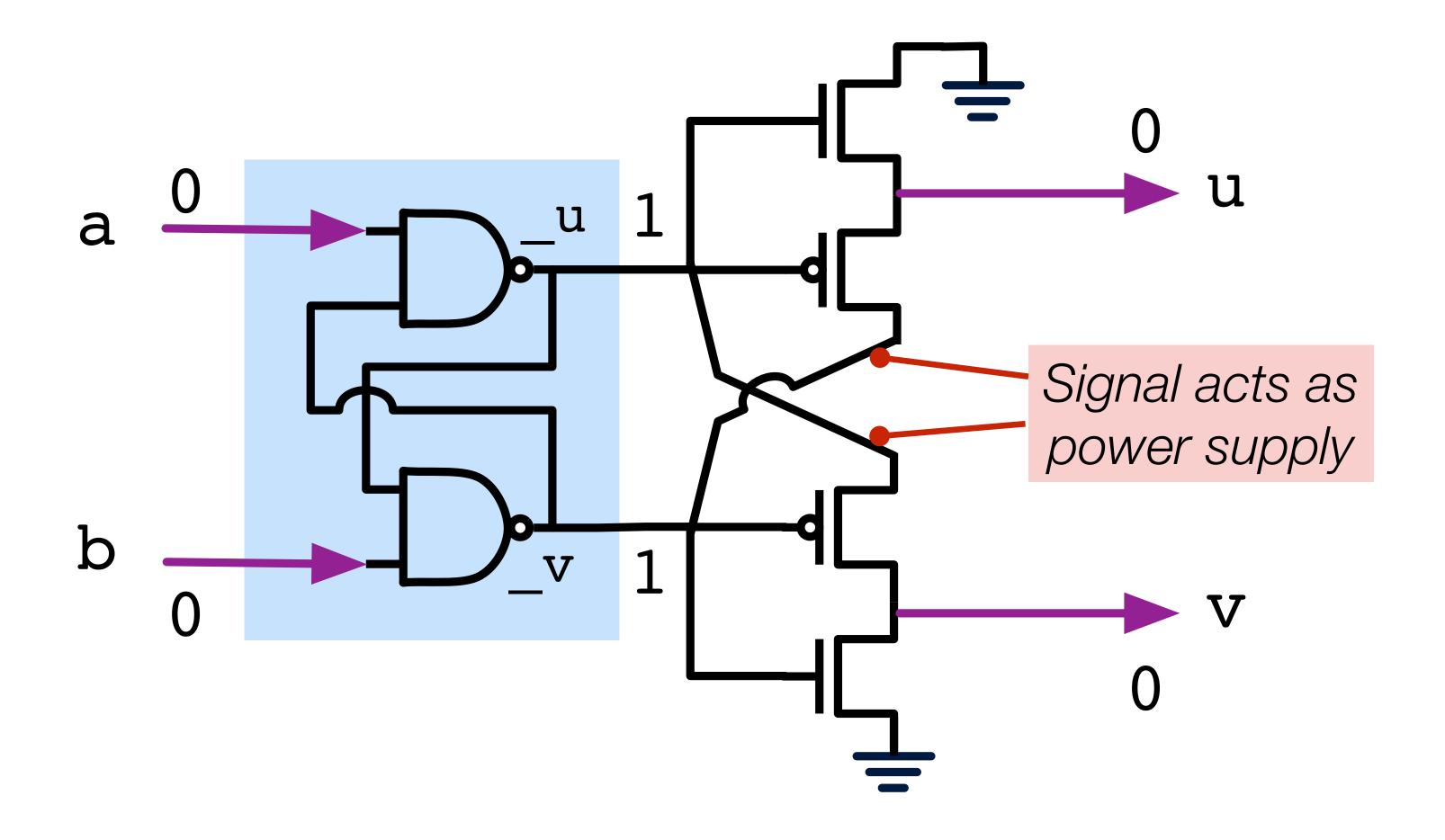




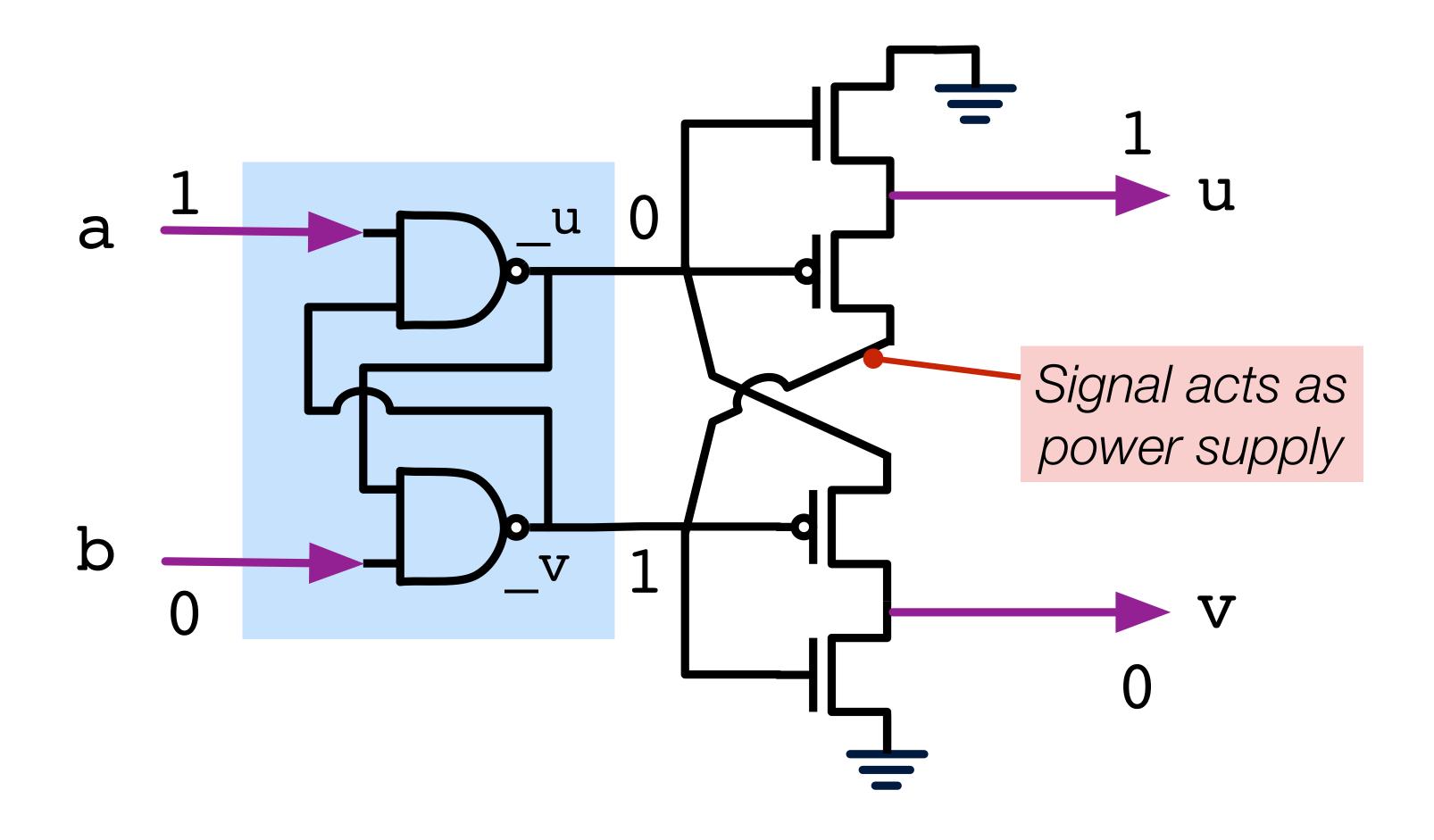




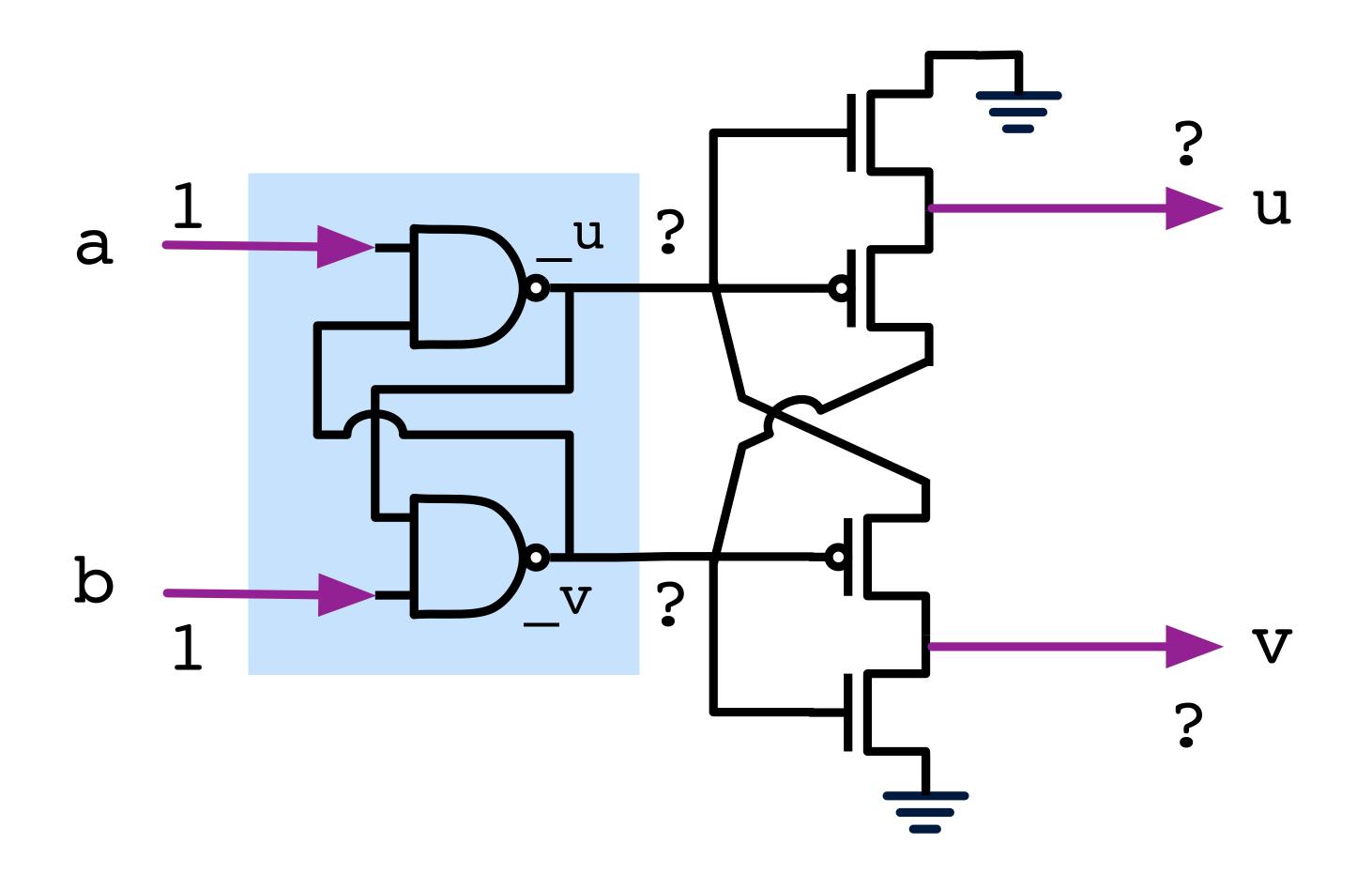






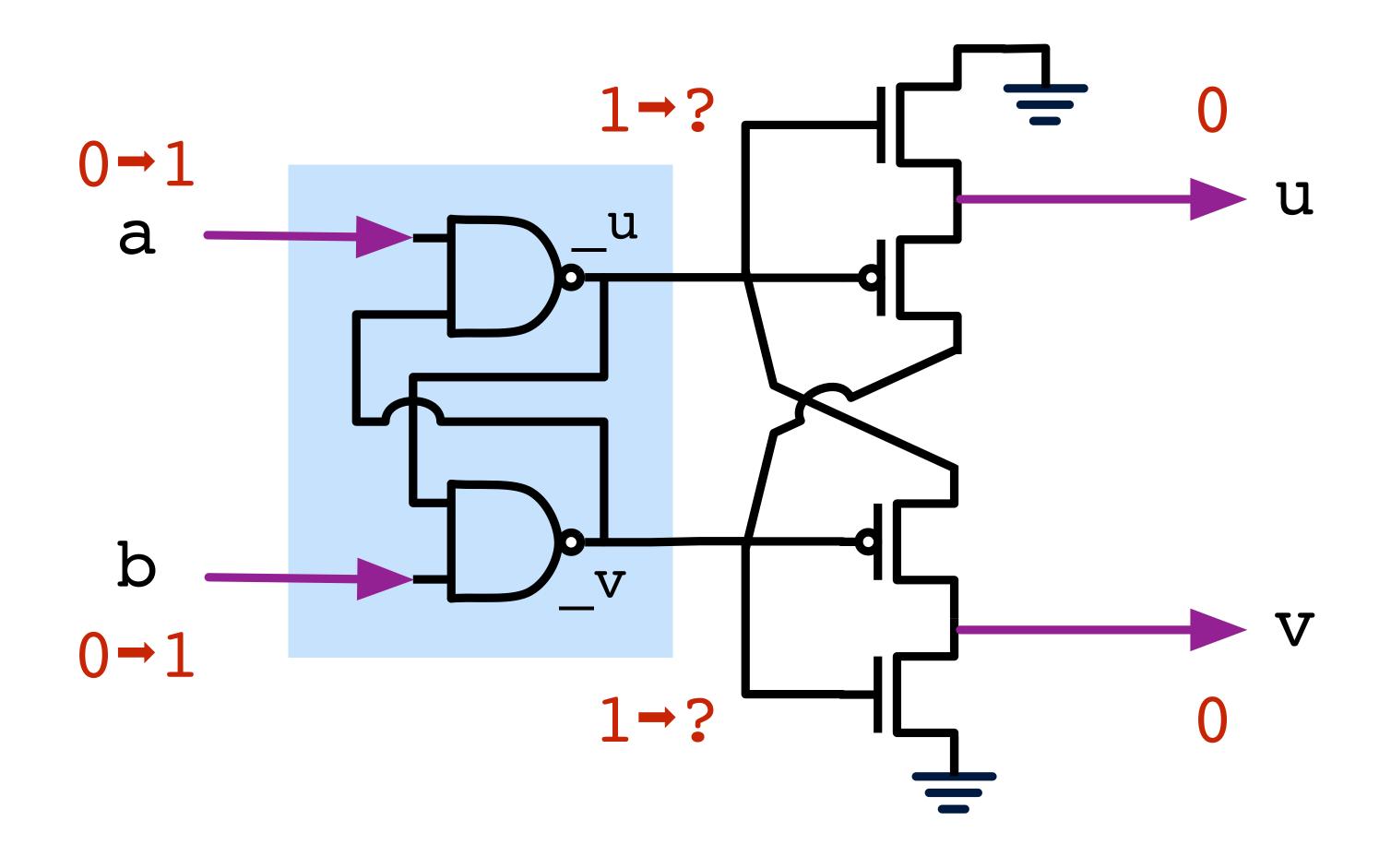






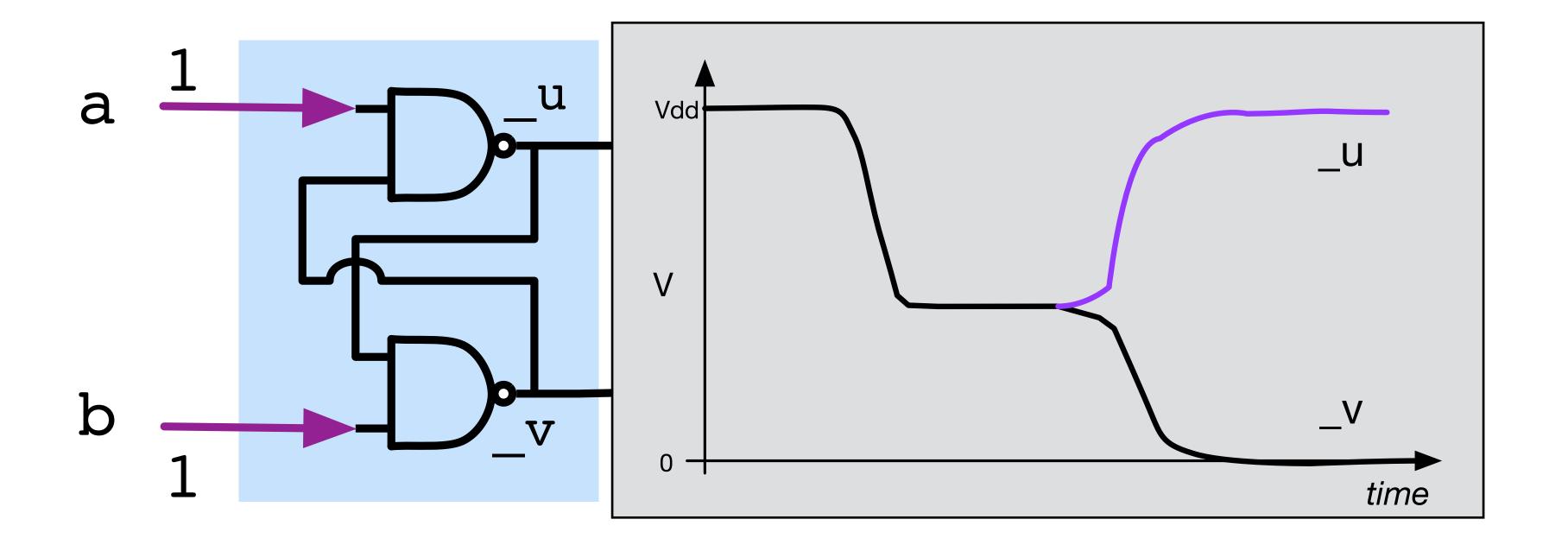
$$\Pr[\text{time} \ge t] = Ae^{-t/\tau}$$





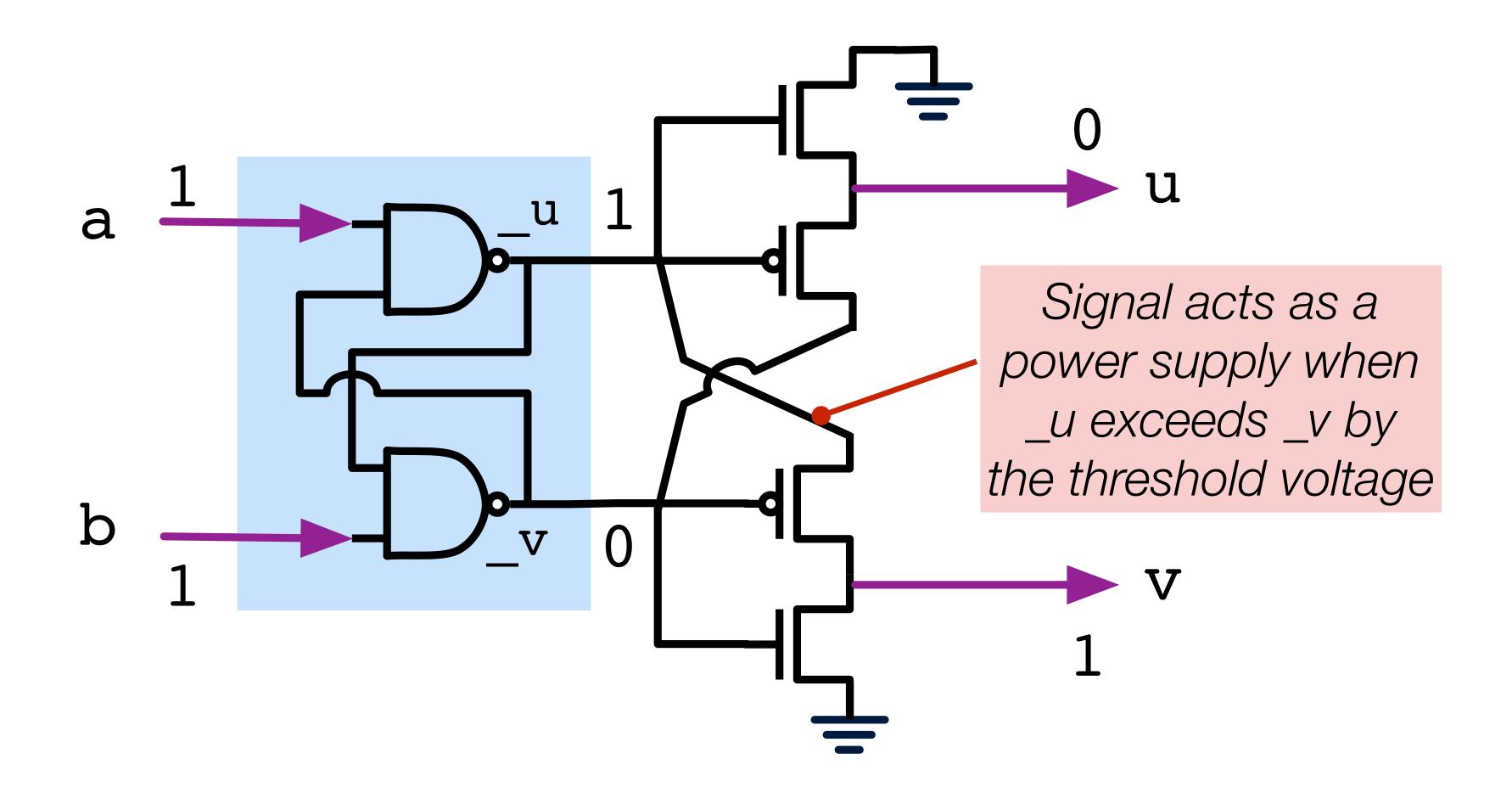
$$\Pr[\text{time} \ge t] = Ae^{-t/\tau}$$





$$\Pr[\text{time} \ge t] = Ae^{-t/\tau}$$





$$\Pr[\text{time} \ge t] = Ae^{-t/\tau}$$

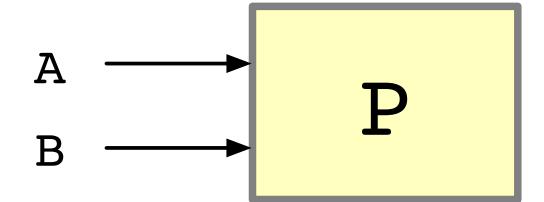


Translating non-deterministic selections

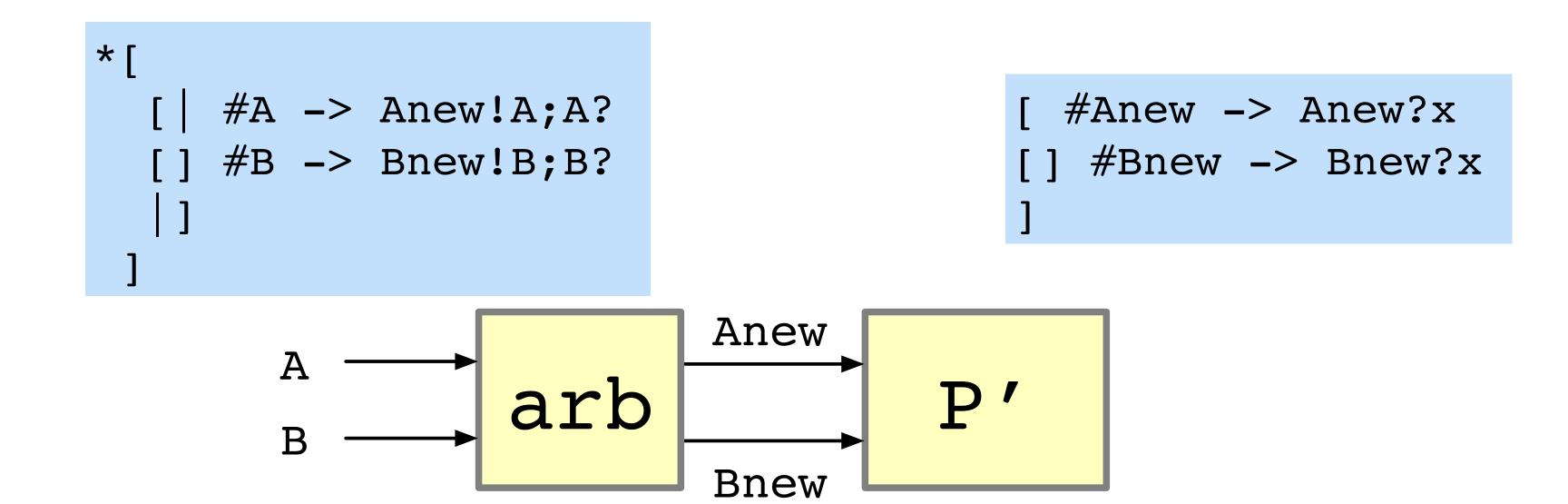
• Basic idea

standard

<u>component</u>



Factor out non-deterministic execution

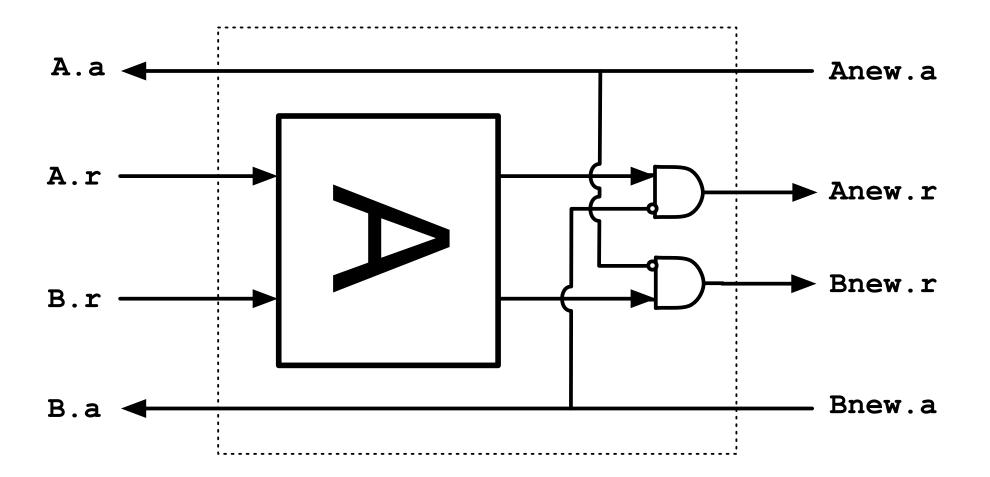






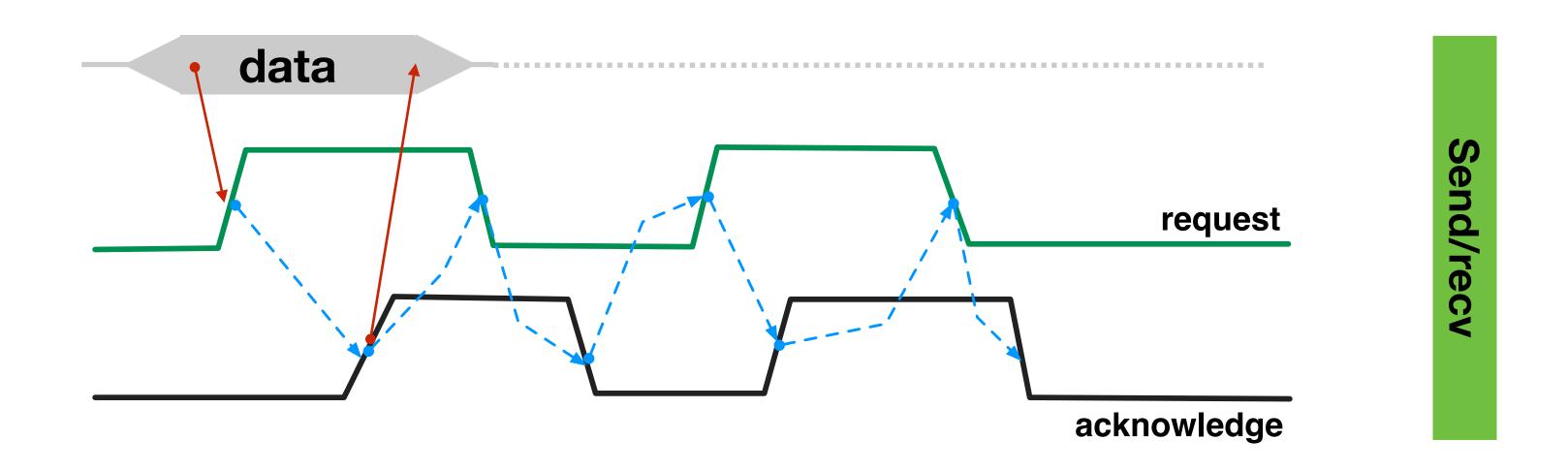
Example circuit

```
*[
    [| #A -> Anew!A;A?
    [] #B -> Bnew!B;B?
    |]
]
```





A simple (slow) interface to clocked environments

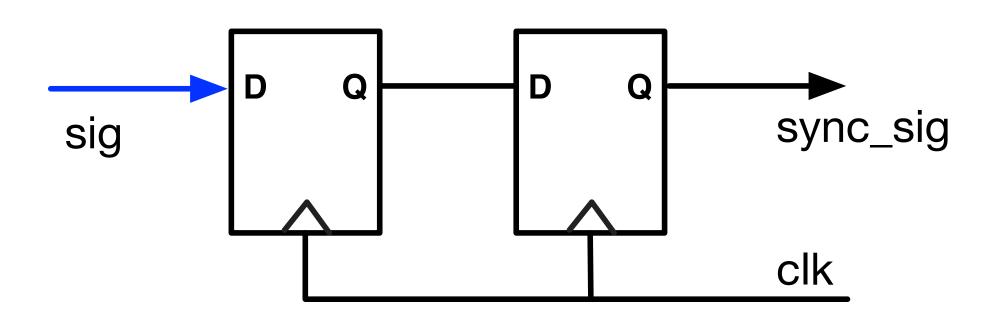


- Send to clocked environment
 - * Request might change when the clock is changing
- Receive from clocked environment
 - * Acknowledge might change when the clock is changing

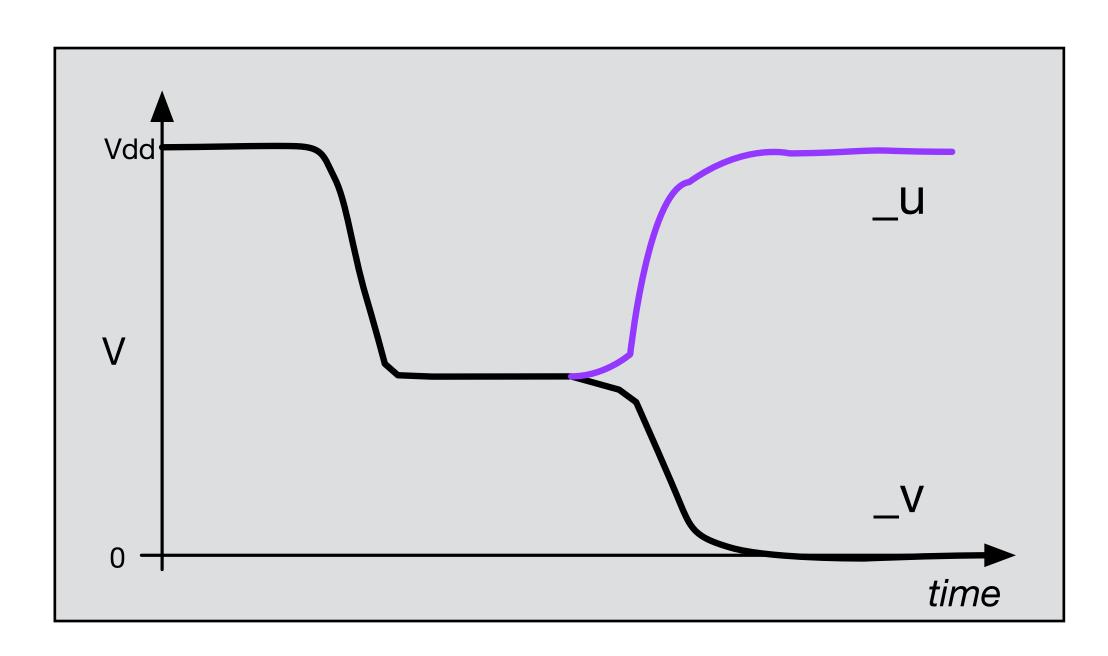




Standard synchronizer: a sequence of flip-flops



- Send to clocked environment
 - * Request might change when the clock is changing
- Receive from clocked environment
 - * Acknowledge might change when the clock is changing



$$\Pr[\text{time} \ge t] = Ae^{-t/\tau}$$



